

## 第十二章 檔案傳輸系統

### 12-1 FTP 系統簡介

『檔案傳輸協定』( **File Transfer Protocol, FTP** ) 也幾乎是和 TCP/IP 網路同時誕生，是 APARNET 網路上最基本且重要的應用系統，基本規範是由 RFC 959 所制定，但目前又有許多新的規範來彌補 FTP 安全性不足，譬如，RFC 2228 ( **FTP Security Extensions** )、RFC 2640 ( **Internationalization of the File Transfer Protocol** )、RFC 2773 ( **Encryption using KEA and SKIPJACK** ) 等等。FTP 主要是應用於檔案傳輸使用，將共享檔案存放於 FTP 伺服器，讓一般使用者可以透過網路來下載或上傳。一般 FTP 訴求的重點是在異質性電腦之間、以及遠距離的檔案共享使用，它和一般區域網路上的『檔案伺服器』( **File Server** ) 有很大的不同點。

常見的檔案伺服器，如 Unix/Linux 系統上的『網路檔案系統』( **Network File System, NFS** )，或者是 Microsoft 網路上的『網路芳鄰』。這些檔案伺服器可以將檔案分享成像是使用者自己電腦上的檔案一般，使用者不但可以複製和刪除分享檔案，並可自由編輯分享檔案(依使用者權限而定)，對使用者而言，存取檔案伺服器上的共享檔案就如同存取自己電腦上的檔案一樣。因此，檔案伺服器是屬於較高階的網路應用，一般只限制於同一網路作業系統(如，Unix/Linux 或 Microsoft)上存取，如跨越不同作業系統的話，可能必須經過特殊的處理(如，Samba 伺服器)。FTP 伺服器則不然，它希望提供一個標準通訊協定，讓不同電腦系統之間可以任意傳輸檔案，來達成檔案共享的目的，雖然它沒有檔案伺服器的高階應用工具，但它可以整合不同系統來互相傳輸檔案，尤其在目前這 Internet 網路時代下，FTP 的應用更顯得重要。

為了能整合異質性電腦之間的通訊，FTP 也是採用最低通用的 NVT ASCII 通訊語法(如第十二章介紹)，將主從式兩端的通訊端點，模擬成如同 Telnet 兩端的命令通訊一樣，對於不同系統之間便可省略掉許多相容性的問題，而這些通訊命令也可用 ASCII 直譯命令來達成(如，USER、STOR 命令)。

標準 Unix/Linux 版本上的 FTP 伺服器，使用者都必須具系統上的使用者名稱及密碼才可以登入。但目前 Internet 網路上共享資源愈來愈多，並且都是無條件讓使用者下載使用，如需要完整系統上的使用者名稱及密碼，將會嚴重限制使用者的方便性，因此，就有『匿名 FTP』( **Anonymous**

FTP) 的誕生。在匿名 FTP 之下，使用者以 anonymous 為使用者名稱，而以電子郵遞帳號作密碼便可登入，來從事檔案傳輸的工作，除非特殊網站，否則一般伺服器並不真正去偵測帳號的真實性，而只判斷是否有『@』來決定是否允許登入。許多廠商將一般共享軟體、或產品更新程式放置於匿名 FTP 網站，讓客戶自由下載。但對於較隱密性資料或軟體，通常限制使用，使用者必須要有帳號及密碼才可登入傳輸檔案。本章首先介紹 FTP 的基本原理，再以目前 Internet 網路上最流行的 wu-ftpd 伺服軟體為範例，來介紹 FTP 伺服器的安裝與管理方法。

## 12-2 FTP 協定

FTP 協定是採用 TCP 連線來製作，應用模式同樣是主從式 ( Client/Server ) 架構，雙方通訊亦使用 Socket 通訊端點 ( 如第十二章介紹 )。

FTP 使用兩個 TCP 連線來傳輸檔案，著名埠口 20 ( tcp/20 ) 做為傳輸資料使用，而另外埠口 21 ( tcp/21 ) 做為傳輸控制訊息使用，以下分別介紹這兩個連線的管理。

### 12-2-1 FTP 控制連線

圖 12-1 為 FTP 建立連線方式，其中包含控制連線與資料連線，控制連線運作如下：

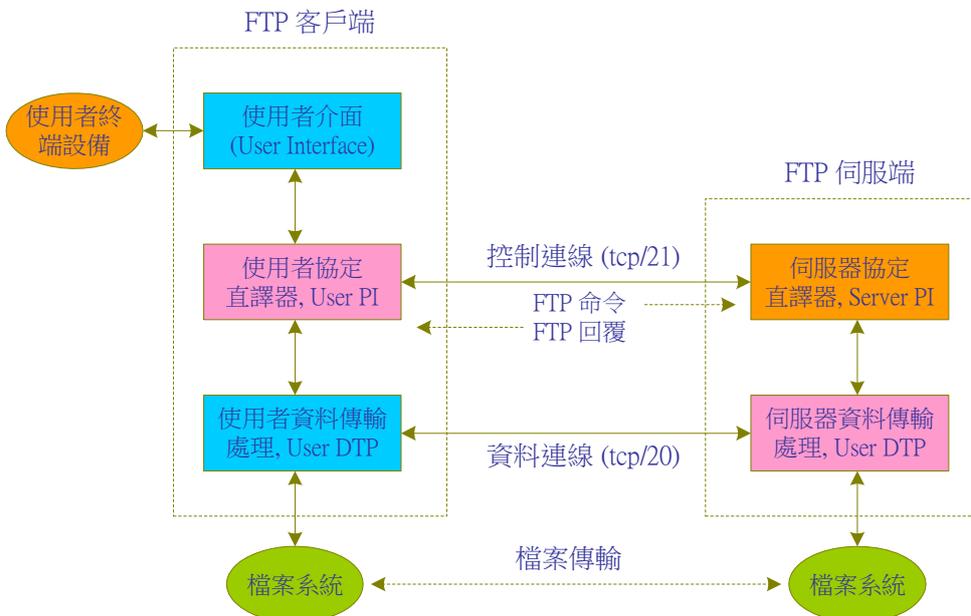


圖 12-1 FTP 連線方式

1. FTP 檔案傳輸系統採用主從式模式，FTP 伺服器隨時監視埠口 21 ( tcp/21 ) 是否有連線要求。FTP 客戶端需要連線時，便由著名埠口 21 連結到 FTP 伺服器。

2. 主從雙方都模擬成 NTV 終端機系統，可由使用者介面輸入 FTP 命令，並將該命令以直譯 ( Interpret ) 方式，編譯成 ASCII 命令，再傳送給對方。因此，主從雙方皆具有『使用者協定直譯器』( User Protocol Interpreter, User PI )和『伺服器端協定直譯器』( Server Protocol Interpreter, Server PI )。
3. 伺服器端同意連線後 ( 驗證使用者名稱及密碼 )，便建立並保持著控制連線，以便隨時交換訊息，當需要傳輸檔案時，再建立資料連線，傳輸後立即釋放該資料連線，一般伺服器端的資料連線都建立在埠口 20 ( 20/tcp )。
4. 雙方透過資料連線來存取各自檔案系統 ( File System ) 中的檔案，此時會牽涉到系統檔案的讀取和寫入動作，而它是由主從式雙方各自呼叫系統函數來達成，該工作是由『使用者資料傳輸程序』( User Data Transfer Process, User DTP )和『伺服器端資料傳輸程序』( Server Data Transfer Process, Server DTR ) 來完成。

目前 Internet 網路上有許多圖形化的『使用者介面』，譬如，Cute-FTP 或 WS-FTP 應用軟體。但一般標準化介面是模擬成有如 Telnet 般的終端機系統，由使用者在鍵盤上輸入文字命令，再經過 User PI 編譯成 FTP 命令傳送給對方。FTP 伺服器軟體的文字命令也透過 Server PI 編譯成 FTP 命令，傳送給客戶端。但雙方的終端機系統並沒有像 Telnet 系統那麼複雜，選項也較為簡單。

### 12-2-2 FTP 資料連線

資料連線是當需要傳輸資料時，才建立而成，當傳輸資料完畢後，該連線就自動消失，下次需要傳輸資料時再建立 ( FTP 控制連線並未中斷 )。資料連線是用來銜接 User DTP 和 Server DTP，其中牽涉到檔案的開啟、讀取、寫入等等有關作業系統本身的系統呼叫 ( System Call ) 的部分，FTP 利用獨立連線來達成，可省略掉許多處理程序，這也是 FTP 和 Telnet 最大的不同點。資料連線有 3 個功能：

- (1) 由 FTP Client 傳送一個檔案到 FTP Server。
- (2) 由 FTP Server 傳送一個檔案到 FTP Client。
- (3) 由 FTP Server 傳送一檔案或目錄清單到 Client 端。

當主從雙方連線後，即一直保持著控制連線，而資料連線則會隨時依需要建立，或關閉 ( 資料傳送完畢或 ABOR 命令中斷 )，又在 FTP 協定上允許任何一方開啟此資料連線，開啟者稱為主動

開啟 ( Active Open ) ; 而接受者稱為被動開啟 ( Passive Open ) , 至於是由誰為主動開啟 ? 這就有協調的必要。

### ( A ) 伺服器主動

一般在主從式的應用系統上，任一事件的發生，大多是由客戶端主動要求，而伺服器才會隨著客戶端需求做出適當的反應，因此是否需要建立的決定權在於客戶端，所以建立資料連線的運作程序如下：

1. 當使用者要求傳輸檔案時 ( 執行 STOR 或 LIST ) , 便由 FTP Client 送出 PORT 命令，指定 FTP Client 端的資料連線之埠口位址。
2. FTP Client 便聆聽該資料埠口是否有連線要求。
3. FTP Server 由控制連線收到 Client 的資料連線埠口位址 ( PORT 命令 ) , 便發出要求連線訊號，FTP Server 的資料埠口大多是固定在 20 號( tcp/20 ) , 但也可能隨需要改變到其它埠口。

圖 12-2 為 FTP Server 主動要求建立資料連線的運作圖，FTP Client 以埠口 1073( 1073/tcp ) 連結到 Server 端的埠口 21( 21/tcp ) , 此為控制連線。但當需要傳輸資料時，FTP Client 發出 PORT 163,15,2,62,4,50 來告訴 Server 端，欲傳輸資料的 IP 位址( 163.15.2.62 )和埠口位址(  $1074 = 4 * 256 + 50$  )。FTP Server 再由埠口 20 ( 20/tcp ) 要求建立資料連線到 Client 端的 1074 埠口。

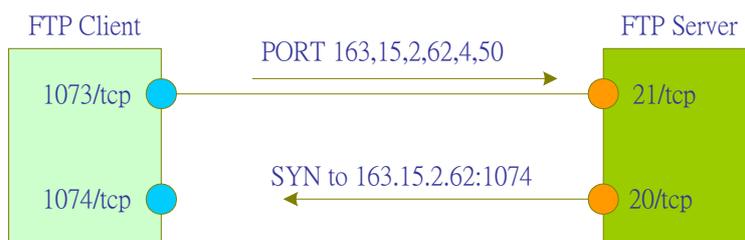


圖 12-2 伺服器主動之運作圖

### ( B ) 客戶端主動

在 FTP 協定的預設是伺服器主動狀態，但有些情況必須由客戶端主動連接，在這種情況下 FTP Client 可利用 PASV( Passive Mode )命令，要求 Server 端進入聆聽狀態等待連接。一般 Server 端的資料連線都建立在埠口 20，因此，Client 端發出 PASV 後，如 Server 端回應執行正常，便可直接要求連線，如圖 12-3 所示。

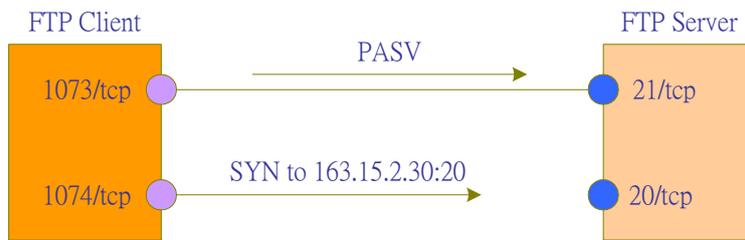


圖 12-3 客戶端主動之運作圖

## 12-3 FTP 傳輸檔案格式

雖然在 RFC 959 中，規範了許多 FTP 傳輸檔案格式的標準，這是希望廠商在製作 FTP 伺服器時，都必須依照這些標準，讓不同電腦系統可以相互傳輸檔案。

但是實作方面，並非所有標準都必須具備，尤其目前 Internet 網路上的 FTP 伺服器大多建立在 Unix 或 Linux 系統上，它所能提供的選項更少。但我們還是依照 RFC 959 上所列的規範來介紹，因為有些地方還是會用到。

### 12-3-1 檔案型態

FTP 協定標準中有下列檔案型態，在製作時至少必須選擇一種：

#### (a) ASCII 檔案型態

ASCII 是一般 FTP 伺服器預設在文字檔案的傳輸中所使用的檔案型態。它也是配合 NTV ASCII 傳輸方法，傳輸之前將檔案內容轉換成 NTV ASCII 格式，到達對方時，再依照 NTV ASCII 格式轉換成本地的資料型態，因此，這種傳輸方式相容性最高。NTV ASCII 是以每 8 個位元為一字元編碼，每一行的結尾用歸位 (Carrier Return, <CR>) 和新行 (Linefeed, <LF>) 組合表示 (如第十二章介紹)。

#### (b) EBCDIC 檔案型態

EBCDIC (Extended Binary Coded Decimal Interchange Code) 是 IBM 電腦上的專屬編碼方式，如果在 IBM 電腦上採用此傳輸格式，可省去許多格式轉換的工作。但在其它電腦系統上，反而增加許多不需要的負擔 (Overhead)，因此，除非 IBM 系列電腦才會選用此傳輸格式，其它電腦系統甚少使用。

### (c) Image 檔案型態

『影像』( Image ) 檔案型態又稱為『二進位』( Binary ) 檔，它是將原來資料儲存的二進位格式，以連續位元流串列傳輸。採用此方法時，檔案都不必經過轉換，直接傳送，一般都應用於傳輸執行檔、影像、聲音等檔案。

### (d) Local 檔案型態

『本地』( Local ) 檔案型態是由傳送端指定傳輸格式，譬如指定本地 36 位元格式，則以 TYPE L 36 命令指定。一般使用於邏輯資料宣告不同的環境，譬如某一端以 36 位元表示浮點數，而另一端以 32 位元表示之。但目前遇到此種情況大多以二進位元檔傳送即可達成。

## 12-3-2 格式控制型態

『格式控制』( Format Control ) 是表示傳輸檔案中，是否包含哪種格式的控制字元，譬如，跳頁，換行等等，一般都跟列印格式較有關係。這個選項只有 ASCII 或 EBCDIC 檔案型態 ( 文字檔案 ) 才有，可選擇下列格式之一：

### (a) 不列印

『不列印』( No Print ) 格式是一般 FTP 協定的預設值，表示在傳輸檔案當中不特殊處理列印控制訊息，如檔案中有列印控制字碼 ( 如，Word 文件 )，也將它看成一般文字型態傳送。

### (b) Telnet 格式控制

檔案內包含針對 Telnet 格式直譯 ( Interpret ) 的列印控制字元，譬如，<CR>、<LF>、<NL>、<VT>、<FF> 與 <CR><LF> 來控制列印格式。

### (c) Fortran Carriage 控制

檔案內包含 Fortran Carriage 的列印控制字元。

## 12-3-3 資料結構型態

『資料結構』( Data Structure ) 表示傳輸檔案的邏輯資料型態，可選擇下列型態之一：

### (a) 檔案結構 ( File Structure )

此為預設型態，它將檔案視為一個連續的位元串流而組成，並沒有內部的檔案結構。

### (b) 紀錄結構 ( Record Structure )

檔案是由紀錄結構所構成，一般使用於傳輸資料庫系統的紀錄檔案，也是 ASCII 和 EBCDIC 檔案型態才具有之結構。一般資料庫的紀錄檔案都非常龐大，依此結構傳輸可減少轉換的工作，但目前大多以二進位檔案型態來傳送。

### (c) 頁結構 ( Page Structure )

檔案裡包含頁數的傳輸，目前幾乎已不使用。

## 12-4 FTP 傳輸模式

在 FTP 協定中，除了必需指定雙方傳輸檔案格式外，還必須指定雙方傳輸模式。傳輸模式是表示資料如何透過資料連線，傳送到對方的方法，而接收端也是要用同樣的傳輸模式來接收資料，FTP 協定有下列三種傳輸模式：

### (1) 串流模式 ( Stream Mode )

此為預設模式，檔案被當成位元組串流來傳輸。如果是傳輸檔案，則以檔案結尾 ( EOF ) 來判斷傳輸資料結束；如果是傳輸紀錄結構檔案的話，就利用一個特殊 2 位元組 ( EOR 與 EOF ) 的序列指示，來判斷傳輸訊息結束。

### (2) 區塊模式 ( Block Mode )

將欲傳輸的檔案以某一位元組數量，分割為若干區塊，每一區塊包含一個區塊標頭和資料區塊，區塊標頭上紀錄該區塊的屬性 ( EOR、EOF、Suspected Error 或 Restart Marker )，與該區塊所承載資料的位元組數量 ( Byte Count )。主從式雙方就以區塊方式來發送與接收。

### (3) 壓縮模式 ( Compressed Mode )

在 RFC 959 規範一個簡單的 run-length 編碼來壓縮連續出現的相同位元組。在文字檔裡，連續出現空白的機率高；而在二進位檔裡，則是 0，可依此特性來壓縮，但目前此壓縮方法已甚少使用。在 12-8-2 節中將介紹目前 Internet 網路上較常用的壓縮方法。

## 12-4-1 FTP 最小傳輸模式

由以上的介紹，可以知道建構一個 FTP 伺服器可依照檔案型態、格式控制、資料結構來決定一個 FTP 的傳輸特性，如經這些選項的選擇，可能超過 70 種以上不同的組合。

由於目前 Internet 網路大多使用 Unix 系統來架設 FTP 伺服器，而 Unix 系統只允許某些選項，所以這些就是目前 Internet 網路上通用的選項，選項如下：

- **檔案型態**：ASCII 或 Image ( Binary )。
- **格式列印型態**：不列印。
- **資料結構型態**：檔案結構。
- **傳輸模式**：串流模式。

## 12-5 FTP 命令與回覆訊息

基本上，Client/Server 雙方是以命令和回覆訊息，來達成雙方通訊。但在 FTP 協定中有『**FTP 命令**』和『**FTP 交談命令**』兩種型態，FTP 命令是傳遞給雙方通訊使用；而 FTP 交談命令則允許使用者下達使用。當使用者由鍵盤輸入 FTP 交談命令後，便由直譯器 ( User 或 Server ) 編譯成 FTP 命令，並且在命令後加入 <CRLF> 控制字元，表示一個命令的結束。首先我們來介紹 FTP 命令，在下一節再介紹 FTP 交談命令。

### 12-5-1 FTP 命令

FTP 命令與回應都是 NTV ASCII 型態，透過主從雙方的控制連線來傳送，任何一方都可發送命令，而接收端收到命令後，再以 FTP 回覆給對方。基本上，FTP 協定也是將通訊雙方模擬成 Telnet 終端機模式，但並沒有像 Telnet 終端機系統那麼複雜，因此，通訊雙方可能出現某些選項命令 ( WILL、WONT、DO 或 DONT )，對方也可能回應 DONT 或 WONT。

每一個 FTP 命令都是以 4 個字母表示，而格式為『**命令 參數 <CRLF>**』，其中命令表示 FTP 命令( 如 USER )、參數為命令中攜帶的參數值( 如 tsnien )，一個完整命令如『**USER tsnien <CRLF>**』，而其中 <CRLF> 為 NTV ASCII 表示一行結束的意思。欲知道有哪些 FTP 命令，可由 Telnet 登

入 FTP 控制埠口 ( 21/tcp )，再以 HELP 命令來查詢，查詢結果如下：( 其中粗體加底線為輸入命令 )

```
[tsnien@linux-1 tsnien]$ telnet 163.15.2.62 21
Trying 163.15.2.62...
Connected to 163.15.2.62.
Escape character is '^]'.
220 linux-1.cu.edu.tw FTP server (Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 20
00) ready.
user anonymous
331 Guest login ok, send your complete e-mail address as password.
pass tsnien@cc.cma.edu.tw
230 Guest login ok, access restrictions apply.
help
212-The following commands are recognized (* =>'s unimplemented).

  USER      PORT      STOR      MSAM*     RNT0      NLST      MKD       CDUP
  PASS      PASV     APPE     MRSQ*     ABOR      SITE     XMKD     XCUP
  ACCT*     TYPE     MLFL*    MRCP*     DELE     SYST     RMD      STOU
  SMNT*     STRU     MAIL*    ALLO     CWD      STAT     XRMD     SIZE
  REIN*     MODE     MSND*    REST     XCWD     HELP     PWD      MDTM
  QUIT     RETR     MSOM*    RNFR     LIST     NOOP     XPWD

214 Direct comments to root@localhost.
```

以上 FTP 命令可分：『存取控制命令』( Access Control Command )、『傳輸參數命令』( Transfer Parameter Command ) 與 『FTP 服務命令』( FTP Service Command ) 等三種，我們將一些較常用的命令說明如下：

### ( A ) 存取控制命令

下列較常用的存取控制命令 ( Access Control Command ) 為 FTP Client 欲連結上 FTP Server 的主要命令：

- ※ **USER**：使用者登入命令，如 `USER anonymous`，表示以使用者名稱 `anonymous` 登入 FTP Server。
- ※ **PASS**：使用者密碼，如 `PASS tsnien@pchome.com.tw`。
- ※ **QUIT**：離開並關閉 FTP 連線。

## (B) 傳輸參數命令

FTP Client 和 Server 之間的傳輸模式，除了預設模式外，也可以利用傳輸參數命令 (Transfer Parameter Command) 來重新設定，常用命令如下：

- **PORT**：由 FTP Client 通知 FTP Server 資料連線的位址，其中包含 IP 位址和 TCP 埠口。譬如，`PORT 163,15,2,62,4,50`，則表示 IP 位址為 `163.15.2.62`，而 TCP 埠口為  $4 * 256 + 50 = 1074$ 。
- **PASV**：(Passive) FTP Client 通知 Server 設定聆聽資料連線狀態。FTP 協定的預設狀態是當控制連線建立後，FTP Client 會立即以 `PORT` 命令告訴 Server 端，自己的資料連線埠口位址，並進入聆聽狀態，等待對方連線 (被動狀態)，而由 FTP Server 主動要求連接資料連線。在某些情況，FTP Client 以 `PASV` 命令通知 Server 進入聆聽狀態 (被動狀態)，而 Client 端成為主動狀態，來要求連接資料連線。命令格式為 `PASV`，不需要參數。
- **TYPE**：設定傳輸資料檔案型態的命令，命令格式為『`TYPE <type code> <format>`』，其中 `type code = A` 為 ASCII；`E` 為 EBCDIC；`I` 為 Image，而 `format = N` 為 Non-print；`T` 為 Telnet Format；`C` 為 Carriage Control。譬如，FTP 協定預設值為 `TYPE AN`。

## (C) FTP 服務命令

FTP 服務命令 (FTP Service Command) 是定義檔案傳輸功能的命令，也包含一些較通用的檔案管理命令，常用命令如下：

- **STOR**：FTP Client 儲存檔案於 Server 端，也就是 Client 端上傳一個檔案到 Server 端，如：『`STOR file1.dat`』。
- **RETR**：FTP Client 向 Server 複製一個檔案，也就是由 Server 端下載一個檔案到 Client 端，如：『`RETE file1.dat`』。

- **STOU** : ( **Store Unique** ) 如同 **STOR** 一樣都是傳輸一個檔案到 Server 端，但 **STOU** 表示傳輸後在 Server 端必須是唯一的檔案名稱，也就是不可覆蓋同一檔名的檔案。
- **REST** : ( **Restart** ) 要求重新啟動傳輸連線。
- **DELE** : ( **Delete** ) 要求刪除 Server 上某一檔案。
- **RMD** : ( **Remove Directory** ) 要求刪除某一檔案目錄。
- **MKD** : ( **Make Directory** ) 要求建立一個目錄。
- **LIST** : 顯示檔案目錄。
- **NOOP** : ( **No Operation** ) 此為 dummy 命令，Server 並未執行任何工作，而回應一個執行正確的訊息。
- **ABOR** : 中斷前一個 FTP 命令及任何資料傳輸。

### 12-5-2 FTP 回覆訊息

FTP Server 執行完 Client 端的命令後，必須將執行結果回應給 FTP Client，此為 FTP 訊息。FTP 訊息是由三個數字來表示執行的結果，並隨後加入訊息字串，三個數字的控制訊息是要給 FTP 軟體看的，而隨後之訊息字串是要給交談式的使用者看的，FTP Client 會依照三個數字的控制訊息給予適當的回應；而訊息字串並沒有嚴格規定訊息格式。FTP 訊息之標準格式如下：

**【xyz <SP> 訊息字串 <CRLF>】**

其中前三個數字 ( xyz ) 表訊息種類，第一個數字 ( x ) 表示命令執行結果是正常 ( Good )、不正常 ( Bad ) 或不完整 ( Incomplete )；第二個數字 ( y ) 表示執行結果的原因；第三個數字 ( z ) 為每一訊息狀態再分類。表 12-1 為訊息狀態第一和第二個數字 ( xy ) 所代表的意義。

**表 12-1 FTP 訊息回覆**

FTP 回覆	回 覆 意 義 說 明
1yz	肯定的已接受處理命令之回覆。亦表示新命令的開始。
2yz	肯定的完成命令回覆，亦表示新命令可以再開始。
3yz	肯定的中間命令回覆，亦表示必須再給予下一個命令補充。

4yz	暫時否定的完成回覆，亦表示命令可以重新再發出。
5yz	永久否定的完成回覆，命令不可再發出重試。
x0z	語法錯誤。
x1z	回應訊息。
x2z	回應處理有關控制或資料連線的命令之訊息。
x3z	回應有關認證和登入計數之訊息。
x4z	未使定使用。
x5z	回應有關處理檔案系統命令的訊息。

我們列出一些常見的典型回覆訊息如下：

- ▲ **125** 資料連線已完成，可開始傳輸資料。
- ▲ **200** 命令執行正常。
- ▲ **214** Help 訊息回覆。
- ▲ **221** 連線終止回覆。
- ▲ **230** 使用者登入，處理完成。
- ▲ **225** 資料連線開啟正常，還未傳輸資料。
- ▲ **250** 要求檔案動作正常，處理完成。
- ▲ **331** 使用者名稱確定，要求輸入密碼。
- ▲ **425** 不能開啟資料連線。
- ▲ **452** 寫入檔案錯誤，儲存空間不足。
- ▲ **500** 語法錯誤 ( 不合法之命令 )。
- ▲ **501** 語法錯誤 ( 參數值無效 )。

基本上，FTP Client 每下一個 FTP 命令，FTP Server 都會回覆一個單行的訊息，如需要回覆多行訊息的話，會在訊息數字後面以『-』表示連結的意思，到最後一行（相同訊息數字）以沒有『-』，來表示訊息結束。

## 12-6 FTP 交談命令

FTP 交談命令是讓使用者和 FTP Server 端之間以交談式模式輸入的命令，它提供較高層次的人機介面，讓使用者更容易操作 FTP 伺服器，譬如檔案刪除、變更目錄、刪除目錄等等交談式命令。一般欲進入 FTP 交談模式，可直接在系統下執行 ftp 命令，然後直接輸入 FTP 交談命令，範例如下：

```
[tsnien@linux-1 tsnien]$ ftp
ftp> ?
Commands may be abbreviated.  Commands are:

!          debug          mdir          sendport     site
$          dir              mget          put           size
account   disconnect      mkdir         pwd           status
append    exit            mls           quit          struct
ascii     form            mode          quote         system
bell      get             modtime      recv          sunique
binary    glob            mput         reget         tenex
bye       hash            newer         rstatus      tick
case      help            nmap         rhelp         trace
cd        idle            nlist        rename        type
cdup     image           ntrans       reset         user
chmod    lcd             open         restart       umask
close    ls              prompt       rmdir         verbose
cr       macdef          passive      runique       ?
delete   mdelete        proxy        send
```

我們大略由上列命令名稱都可知道它的功能為何，但是我們還是將較常用的命令條列如表 12-2。

**表 12-2 常用之 FTP 交談命令**

命令名稱	功 能 說 明
?	顯示命令 ( help )。
!	執行本機命令，如 !ls 表示執行本機上的 ls 命令。
ascii	設定 ASCII 模式傳輸，使用於文字檔案傳輸。
binary	設定二進位模式傳輸，一般使用於執行檔或影像檔傳輸。
bye	結束 FTP 連線。
cd	切換伺服器的工作目錄。
close	關閉 FTP 連線。
debug	進入偵錯模式。
delete	刪除遠端伺服器檔案。
dir	顯示遠端伺服器工作目錄下的檔案和目錄清單。
get	下載一個檔案
mdelete	刪除多個遠端檔案。
mget	下載多個檔案。
mkdir	在遠端伺服器建立目錄。
mput	上傳多個檔案到遠端伺服器。
open	開啟 FTP 連線。
put	上傳一個檔案到遠端伺服器。
pwd	顯示目前遠端伺服器的工作目錄。
rmdir	刪除遠端伺服器的目錄。
status	顯示目前連線狀態。
type	設定檔案傳輸模式。

user	登入 FTP 伺服器使用者名稱 ( 如 user tsnien )。
------	------------------------------------

如果要觀察雙方傳輸的運作情形，可以用 `debug` 命令，使雙方運作進入 `Debug` 模式。我們開啟一個 `FTP` 連線並上傳一個檔案到伺服器端作為範例，來觀察雙方運作的情形如下：( 粗體加底線為輸入命令 )

```
[tsnien@linux-1 tsnien]$ ftp
ftp> debug
Debugging on (debug=1).
ftp> open 163.15.2.30
Connected to 163.15.2.30.
220 linux-2.cu.edu.tw FTP server (Version wu-2.6.1-18) ready.
Name (163.15.2.30:tsnien): tsnien
---> USER tsnien
331 Password required for tsnien.
Password:
---> PASS XXXX
230 User tsnien logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put test1.c
local: test1.c remote: test1.c
---> TYPE I
200 Type set to I.
ftp: setsockopt (ignored): Permission denied
---> PORT 163,15,2,62,4,3
200 PORT command successful.
---> STOR test1.c
```

```
150 Opening BINARY mode data connection for test1.c.  
226 Transfer complete.  
43 bytes sent in 0.0158 secs (2.7 Kbytes/sec)  
ftp>
```

由以上我們可以觀察出，FTP 交談命令和 FTP 命令的轉換過程，以及命令執行後的 FTP 訊息回覆（如 220、200）（如果沒有執行 debug 就無法完全觀察到）。譬如，登入時轉換成 USER 命令；要求密碼換成 PASS 命令；執行傳輸檔案命令（put）後，客戶端送出 TYPE I 設定傳輸模式、PORT 指定資料連線位址，並進入聆聽狀態，再執行傳輸命令（STOR）。

## 習題

1. 請說明『FTP 伺服器』（FTP Server）和一般『檔案伺服器』（File Server）有何異同？
2. 請說明 FTP 協定的運作程序。
3. 為何 FTP 傳輸系統必須建立兩條連線？請分別說明建立時機及其功能。
4. 請說明『PSAV』命令的功能。
5. 請說明『PORT』命令的功能，並說明 IP 位址和埠口號碼的表示方法。
6. 請說明一般 FTP 系統傳輸中的最小模式。
7. 請在電腦上（Windows 98）執行 ftp 命令連結到 ftp.nsysu.edu.tw，下載一個檔案，或由 IE 來下載同一檔案，有何不同？
8. 請設定您的 FTP Server，當使用者以 anonymous 登入時，會出現本伺服器的說明和歡迎詞。
9. 同上題，請準備某些檔案讓 anonymous 使用者登入後可以下載，並利用 IE 測試其結果。
10. 請利用 ftp 命令進入 FTP 伺服器的主機（如，ftp 163.15.2.30），請擷取（請參考附錄 A）該命令之建立連線的三向式連結的 TCP 封包，並說明它們之間訊號（Sequence Number、Acknowledge Number、SYN、ACK）傳遞的情形。
11. 同上題，請擷取下達使用者名稱（User:）和 FTP 回覆訊息的封包，並說明其內容。

12. 同上題，請擷取下達使用者密碼 ( Password: ) 和 FTP 回覆訊息的封包，並說明其內容及安全性問題。
13. 同上題，請執行 PORT 命令來設定資料連線埠口，請擷取該命令和 FTP 回覆訊息封包，並說明其內容。
14. 同上題，請上傳一個檔案到 FTP Server，來擷取傳送與回覆封包，並說明其內容。
15. 同上題，請執行 bye 命令 ( ftp>bye )，來擷取 TCP 連線斷線的封包 ( 三向式連絡 )。